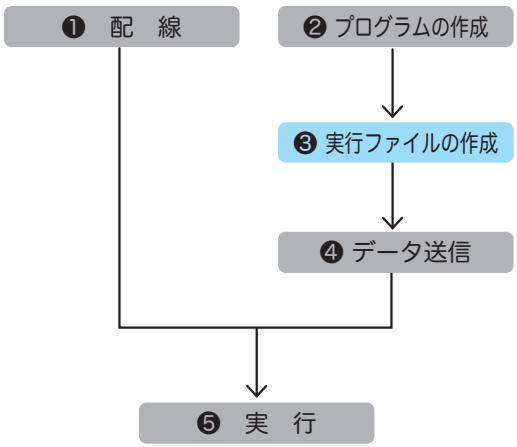
STEP 03

実行ファイルの作成、送信、実行

この STEP では、実際にドットマトリクス LED を点灯させます。
STEP02 で作成したプログラムをマイコン上で実行してみましょう。
プログラムを実行するには、まず実行ファイルを作成し、そのファイルをマイコンに書き込む必要があります。

3.1　make と「Makefile」



STEP02 では、C 言語でプログラムを作成しました。プログラムを組んでいて感じたかと思いますが、C 言語は英語に似ていたと思います。つまり、人間に分かりやすいプログラム言語なのですが、マイコンは C 言語をそのまま実行することはできません。

また、STEP02 で作成した C 言語のソースファイルだけでプログラムを実行することもできません。プログラムを実行するには、他にも必要なファイルがあるのですが、それらは開発環境構築の際にパソコンにコピーした「exercise」フォルダの中に含まれています。

make によって、STEP02 で作成したソースファイルと必要なファイルをつなぎ合わせ、実行ファイル (マイコンへ送信するファイル) を生成します。

STEP 03 実行ファイルの作成、送信、実行

実習でプログラムを作成し、動作させるまでの流れを大雑把に図示すると以下のようになります。作業にとりかかる前に、このような一連の流れを頭の中でイメージしておくといいでしょう。

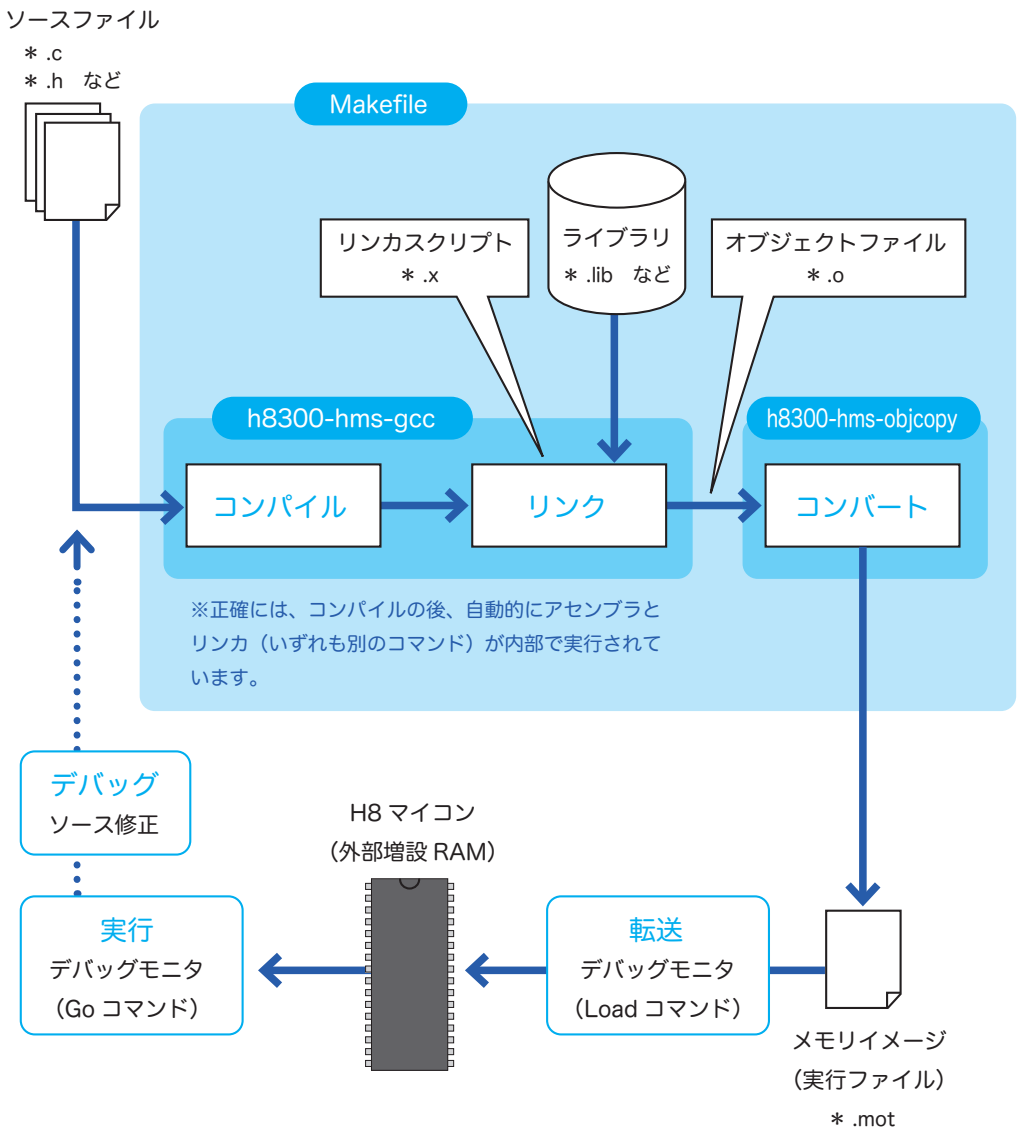


図 3-1　プログラム実行の流れ

実行ファイルは WSL（Ubuntu）で make を実行して生成します。ですが、make を行うには、「Makefile」というファイルが必要です。「Makefile」には、make に必要な情報が記述されています。
自分で「Makefile」を編集・作成することもできますが、STEP02-1 に合わせた「Makefile」が既に用意されています。exercise＞STEP02-1 の中にある Makefile をソースコードエディタで開いてみましょう。

```
CC   = h8300-hms-gcc
OBJCOPY   = h8300-hms-objcopy

SOURCES   = step2-1.c     ･･･①

STARTUP   = crt0.S
EXIT      = exit.c
OBJS      = $(STARTUP:.S=.o) $(EXIT:.c=.o) $(SOURCES:.c=.o)

TARGET       = step2-1     ･･･②
TARGET_COFF  = $(TARGET).coff
TARGET_MOT   = $(TARGET).mot
TARGET_MAP   = $(TARGET).map
    ⋮
```

図 3-1 「Makefile」(一部抜粋)

図 3-1 の青文字部分は、作成したソースファイルの名前 (図中 ①) と実行ファイルの名前 (図中 ②) を記します。

実行ファイルの名前は自由に付けられますが、ソースファイルと実行ファイルの名前が違うと分かりにくくなるので、同じ名前を付けるといいでしょう。
STEP02 で作成したソースファイルは「step02-1.c」なので、このままで問題ありませんが、ソースファイルを別名にした場合は Makefile 内の①を書き換える必要があります。

Makefile 内の他のコマンドについては、本テキストでは説明を省きます。興味のある方は調べてみてください。では、make を実行してみましょう。

3.2 実行ファイルの作成

make は WSL（Ubuntu）で行いますので、まずは WSL を起動してください。
make は、make したいソースファイルと「Makefile」のあるディレクトリで行います。
この STEP では、ディレクトリ「STEP02-1」がそれに当たります。Ubuntu でディレクトリ「STEP02-1」に移動しましょう。
C ドライブ直下に exercise フォルダをコピーした場合は、以下のようにコマンドを実行します。

```
$ cd /mnt/c/exercise/STEP02-1
```

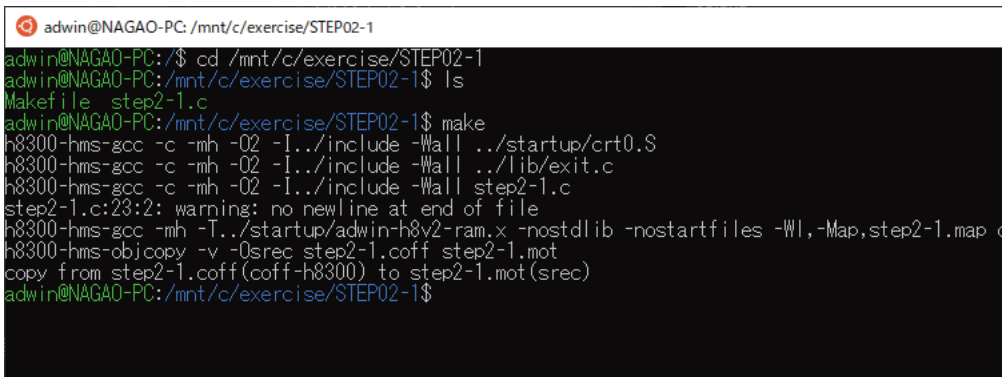
ls コマンドで、ソースファイルと Makefile があるか確認しておきましょう。

```
$ ls
```

ソースファイルと Makefile があることが確認できれば、make コマンドを実行します。

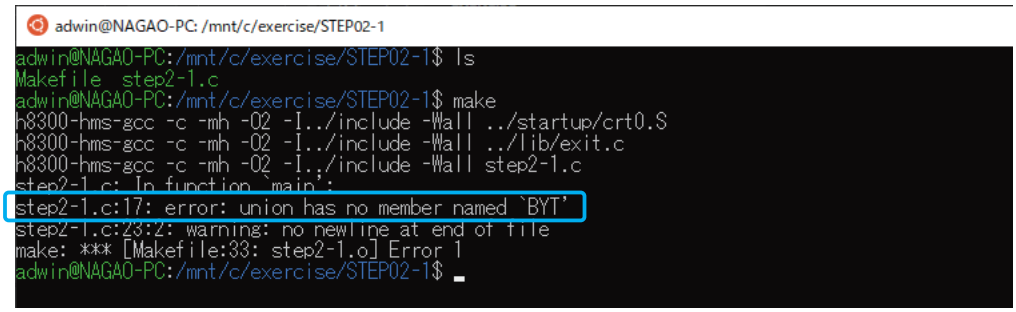
```
$ make
```

下図のように表示されれば、make が成功したことになります。下図のように表示されなかった場合は、STEP02 で作成したソースファイルと「Makefile」を見直してください。



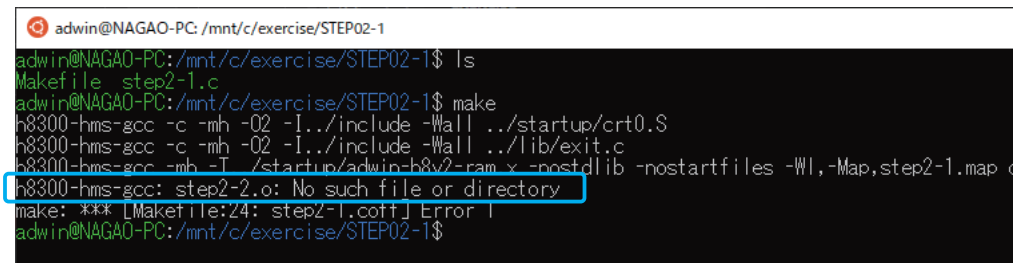
エラーが出たら、ソースファイルの記述ミスや Makefile の内容に記述ミスがあると考えられます。エラーメッセージをよく見て原因を突き止めましょう。

下図の四角で囲まれた部分を見てください。



error 表示の左側の数字は、エラーのある行を示しており、上図の場合だとソースプログラムの 17 行目にエラーがあることになります。
error の右側に示された内容を見ると、「BYT」がヘッダファイルにもソースプログラムにも定義されていない文字なのでエラーになったようです。このエラーの場合、ソースファイル 17 行目の BYT を BYTE と修正すれば解決します。

では、次のようなエラーはどうでしょうか？



このエラーは、エラーのある行数が表示されていません。四角で囲まれた部分を見ると、「step2-2.o ファイルが見つかりません」という意味です。.o ファイルは make 時に生成されるオブジェクトファイルなので、Makefile を開き、ファイル名の指定、フォルダの位置などを確認してください。このエラーは、Makefile 内で指定したソースファイル名と実際のソースファイル名の不一致によるものでした。

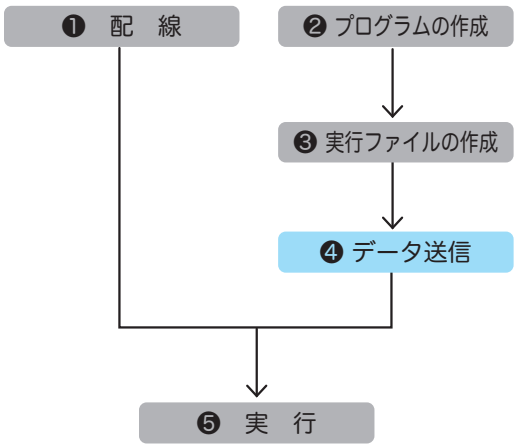
STEP 03 実行ファイルの作成、送信、実行

さて、これで実行ファイルが作成できました。
make を行うと、「STEP02-1」ディレクトリの中に複数のファイルが生成されます。
マイコンへ送信する実行ファイルは、「mot（モト）ファイル」と呼ばれる拡張子が「.mot」のファイルです。
拡張子を表示するように設定しておいてください。拡張子なしでは STEP02-1 というファイルがいくつもあって区別しにくい場合があります。

ソースファイルを修正し、再度 make するときには、一旦、出力結果（中間出力結果を含む）を全て消去しておくようにしましょう。
make clean を実行すると、make で生成されたファイルは削除することができます。.c ソースファイルと Makefile だけになります。

```
$ make clean
```

3.3 データ送信

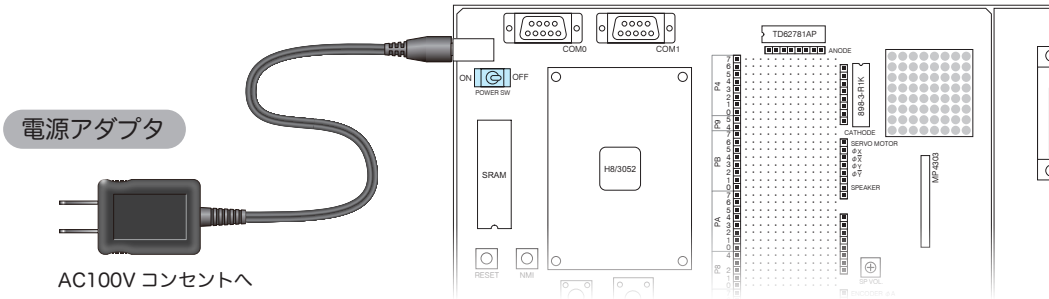


次は、make によって作成された MOT ファイル (実行ファイル) をマイコンへ送信しましょう。
H8 マイコンに実行ファイルを送信するには、「Tera Term」を使用します。

まずは、実験キット本体と電源をつなぎ、キットとパソコンを接続しましょう。

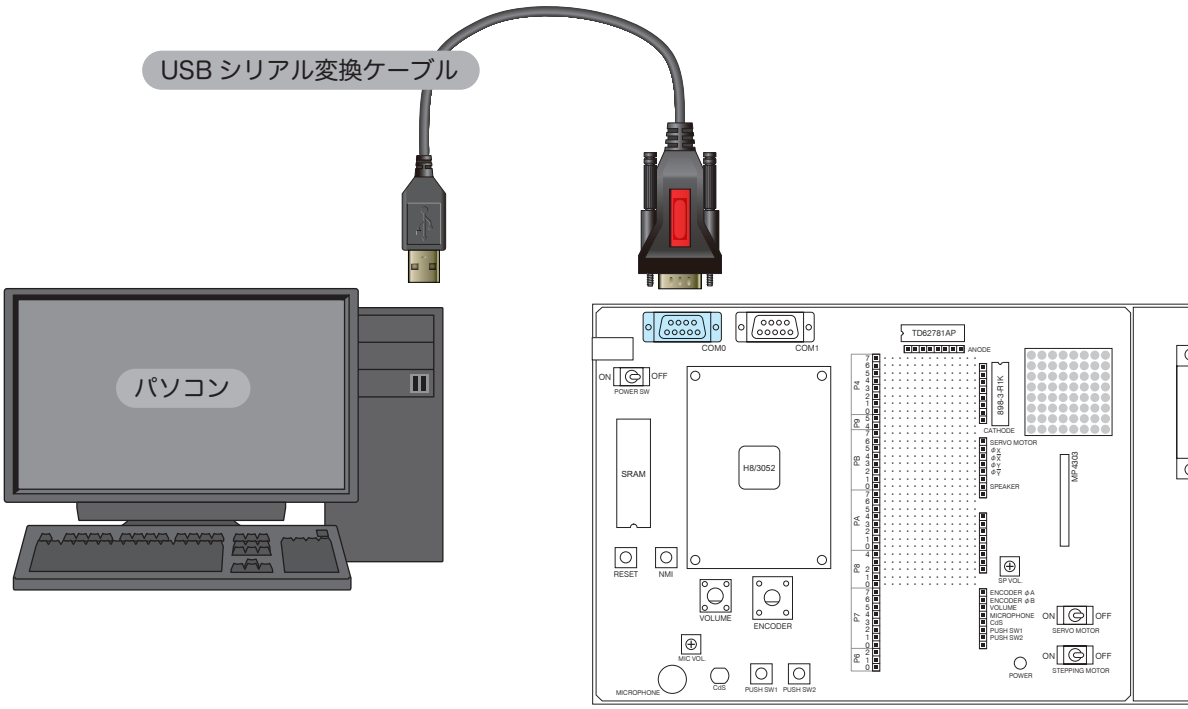
① 実験キットと電源をつなぐ

付属の「電源アダプタ」を接続し、キット本体に電源を供給します。まだ、キット本体の主電源スイッチは OFF にしておいてください。



② 実験キットとパソコンを接続する

付属の「USB シリアル変換ケーブル」で、キット本体のシリアル通信用コネクタ COM0 とパソコンの USB ポートを接続してください。

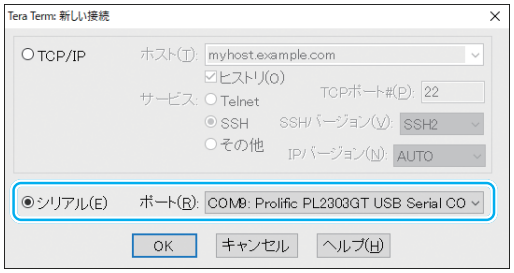


③ シリアルポートを確認する

「開発環境構築」の「USB ドライバのインストール」時に確認した COM 番号は分かるでしょうか。不明の場合はデバイスマネージャを開き、再度確認しておいてください。

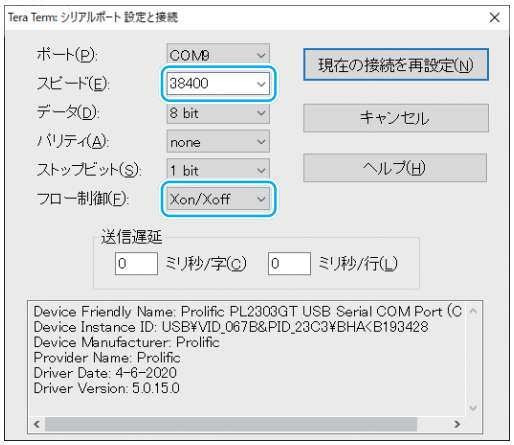
④「Tera Term」を起動する

データ送信は「Tera Term」を使って行います。
まずは、「Tera Term」を起動し、以下の手順で「Tera Term」の設定を行ってください。



❶ 起動時に表示されるダイアログで、シリアルポートを選択し、③で調べた COM 番号をプルダウンメニューから選択し、「OK」をクリック。

❷「Tera Term」の起動後、メニューの 設定 > シリアルポート を選択。



❸ シリアルポート設定で

スピード (E)： 38400
データ (D)： 8 bit
パリティ (A)： none
ストップ ビット (S)： 1 bit
フロー制御 (F)： Xon/Xoff

と設定して、「現在の接続を再設定」をクリック。

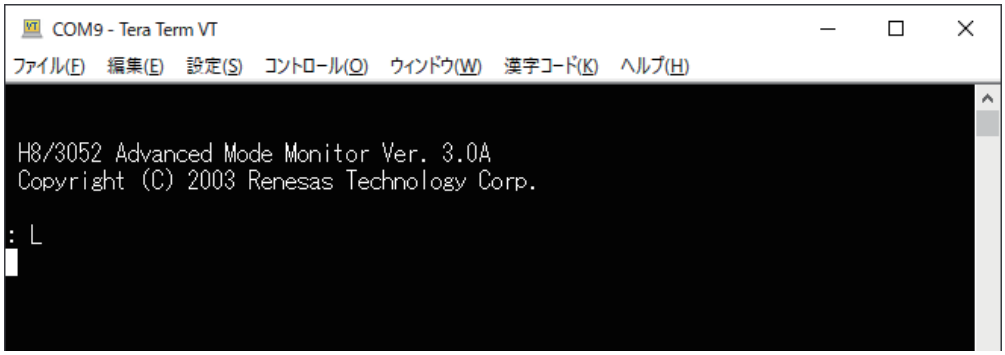
❹ メニューの 設定 > 設定の保存 を選択し、TERATERM.INI ファイルを ttermpro.exe と同じディレクトリに上書き保存しておくと、次回起動時から上記の設定は不要になります。

「Tera Term」の画面が表示された状態で、実験キットの電源を入れてください。
キット本体の電源ランプが点灯し、「Tera Term」の画面に下のようなメッセージが表示されます。これで、「Tera Term」の設定が完了したことになります。下図のような画面が表示されない場合は、ケーブルの接続、ポートの設定、電源が入っているかどうかを確認してください。

⑤ データを送信する

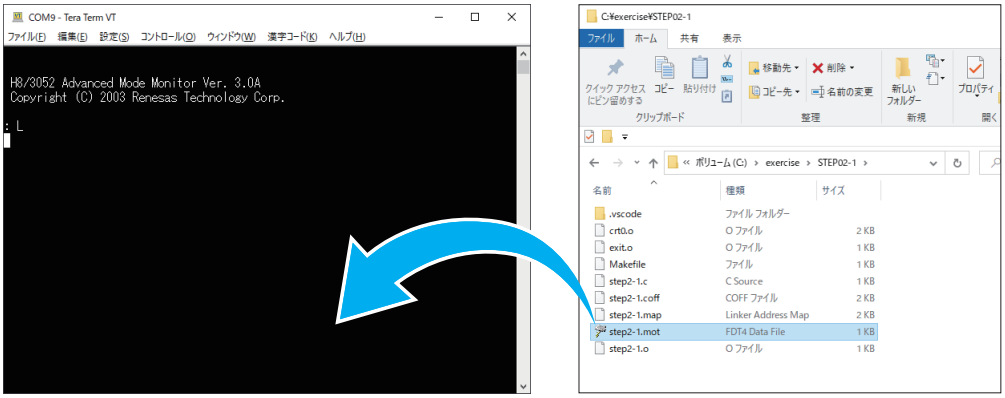
H8 マイコンに実行ファイルを送信します。
下記の手順に従って、H8 マイコンに実行ファイルを送信してください。

❶ 実行ファイルの送信は、L コマンドで送信データ待機状態になります。L は Laod の略で、小文字、大文字どちらでも構いません。「L」と入力して Enter キーを押します。

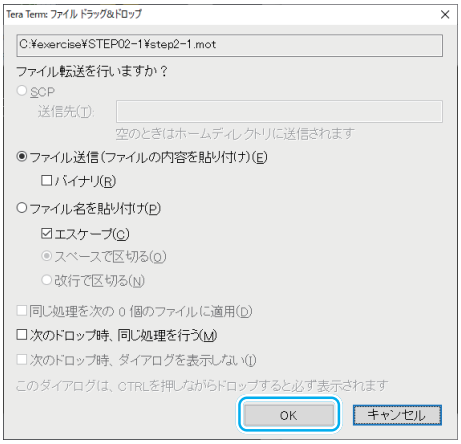


STEP 03 実行ファイルの作成、送信、実行

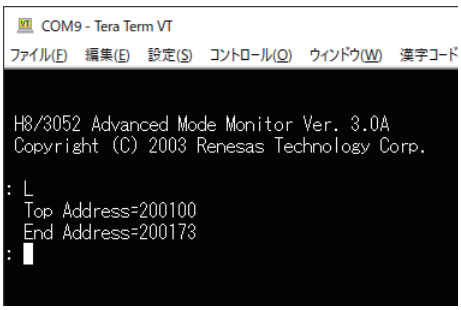
❷ 送信データ待機状態になった Tera Term ウィンドウに、mot ファイル（step02-1.mot）をドラッグ & ドロップします。
メニューで ファイル ＞ ファイル転送 を選択し、ファイル送信ダイアログで mot ファイルを選択する方法でも指定できます。



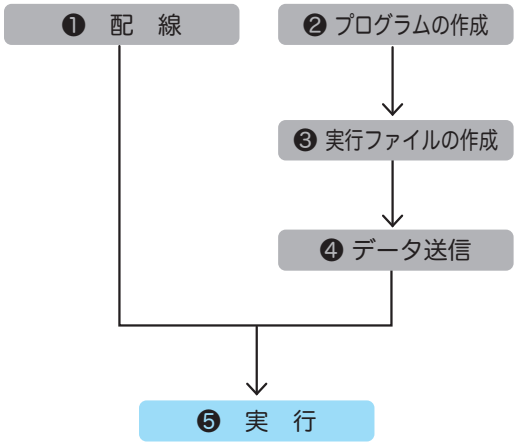
❸ mot ファイルをドラッグ & ドロップした場合は、左のようなダイアログが表示されますが、そのまま「OK」をクリックします。



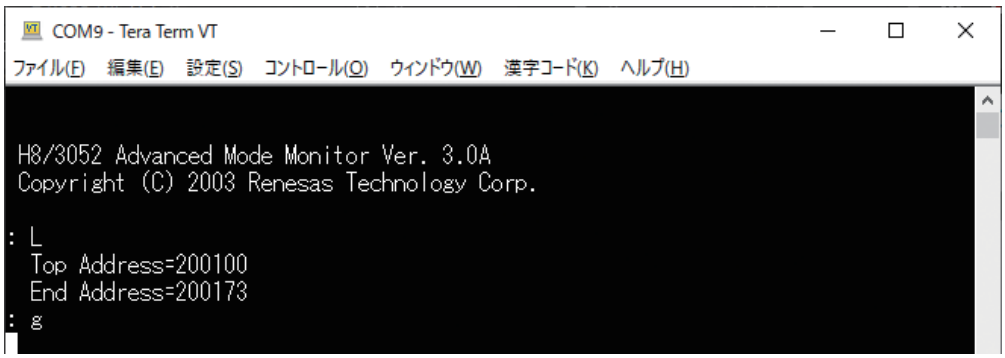
❹ mot ファイルが H8 マイコンに送信され、左図のように、表示されればデータ転送の完了です。
End Address は mot ファイルのサイズによって変化します。



⑥ 実行



H8 マイコンへ送信したファイルを実行は、Gコマンドで開始します。
G は Go の略で、L と同様に小文字、大文字どちらでも構いません。
「G」と入力して Enter キーを押します。



H8 マイコンキットのドットマトリクス LED が右上だけ点灯すれば成功です。
「make」、「データ送信」、「実行」の作業手順は、作成したソースファイルを実行しようとするたびに必要となりますのでよく覚えておいてください。

リセットボタンの使い方

以下の場合は、キット本体の電源を切るのではなく「リセットボタン」を押して対応してください。

・コマンドを間違えて入力してしまった場合
・マイコンに転送する実行ファイルの選択を間違えてしまった場合
・一度実行ファイルを転送した後、別の実行ファイルを転送したい場合

STEP 03 実行ファイルの作成、送信、実行

マイコンに電源を入れたとたん以下のようなメッセージが表示されるのは、本キットのマイコンの ROM に予め「デバッグモニタ」と呼ばれるソフトウェアが書き込まれているためです。ROM に書き込んであるので電源を切っても消えることはありません。
デバッグモニタについて詳しくは「リファレンス .pdf」で解説しています。